UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/721,695	11/24/2000	Matthew A. Feinberg	C24-002	4384

28156 7590 08/23/2005

COLEMAN SUDOL SAPONE, P.C.
714 COLORADO AVENUE
BRIDGE PORT, CT 06605-1601

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2195	

DATE MAILED: 08/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/721,695	FEINBERG, MATTHEW A.	
	Examiner	Art Unit	
	Kenneth Tang	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 March 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *21-30* is/are allowed.

6)☒ Claim(s) *1-3,5-13,15-20,31-33 and 35-45* is/are rejected.

7)☒ Claim(s) *4,14 and 34* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the Amendment filed on 3/2/05. Applicant's arguments have been fully considered but they were not found to be persuasive.

2. Claims 1-45 are presented for examination.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 41 recites the limitation "the computer" in lines 16-17. There is insufficient antecedent basis for this limitation in the claim. The Examiner recommends amending "the computer" to "the multi-tasking computer".

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-3, 5-13, 15-18, 20, 31-33, 35-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Diepstraten et al. (hereinafter Diepstraten) (US 6,260,150 B1).

5. As to claim 1, Bitar teaches a method for operating a computer, comprising:

storing in a computer memory a plurality of pseudocode instructions, at least some of said pseudocode instructions comprising a plurality of machine code instructions *(col. 6, lines 9-12 and 43-65, col. 5, lines 34-41)*;

for each of a plurality of tasks or jobs to be performed by the computer, automatically creating a respective virtual thread of execution context data including (a) a memory location of a next one of said pseudocode instructions (thread to run in next time slice) to be executed in carrying out the respective task or job (mapped) and (b) the values of any local variables required for carrying out the respective task or job, a plurality of said tasks or jobs each entailing execution of a respective one of paid pseudocode instructions comprising a plurality of machine language instructions; processing tasks or jobs in a respective series of time slices or processing slots under the control of the respective virtual thread; and context switching between threads *(col. 1, lines 25-35, col. 4, lines 10-20 and 57-67, col. 5, lines 19-25, col. 6, lines 9-12, col. 13, lines 30-49)*.

6. Bitar fails to explicitly teach performing context switching only after completed execution of a currently executing one of said pseudocode instructions. However, Diepstarten teaches time slice context switching to occur at the end of every instruction *(col. 8, lines 42-49 and 63-66)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of time slice context switching to occur at the end of every instruction to the existing system of Bitar because it is preferred to switch states or instructions when the instruction is over/completed *(col. 8, lines 50-66)*.

7. As to claim 2, Bitar teaches wherein each of the virtual threads is part of a respective linked list of virtual threads, each of the virtual threads further including a pointer to a next virtual thread (thread to run in next time slice) in the respective linked list, further comprising, for every context switch between different virtual threads, consulting the pointer of a currently executing virtual thread to determine an identity of a next virtual thread to be executed *(col. 13, lines 30-49, col. 16, lines 5-7)*.

8. As to claim 3, Bitar teaches wherein said respective linked list is one of a plurality of linked lists of said virtual threads, one of said linked lists being a list of idle virtual threads, another of said linked lists being a list of active virtual threads, an additional one of said linked lists being a list of queued virtual threads, further comprising periodically moving at least one virtual thread from said list of queued virtual threads to said list of active virtual threads *(col. 11, lines 1-7, col. 12, lines 7-21)*.

9. As to claim 5, it is rejected for the same reasons as stated in the rejection of claim 4.

10. As to claim 6, it is rejected for the same reasons as stated in the rejection of claim 4.

11. As to claim 7, it is rejected for the same reasons as stated in the rejection of claim 2.

12. As to claim 8, Bitar teaches wherein each of said virtual threads is assigned a message queue, further comprising entering a message in a message queue of a selected one of said virtual

threads during execution of a task or job pursuant to another one of said virtual threads *(col. 17, lines 22-35 and col. 18, lines 21-41)*.

13. As to claim 9, Bitar teaches wherein said selected one of said virtual threads and said another one of said virtual threads correspond to respective tasks or jobs derived from different applications programs, whereby the entering of said message in the message queue of said selected one of said virtual threads implements data transfer between said different applications programs *(col. 17, lines 22-35 and col. 18, lines 21-41)*.

14. As to claim 10, Bitar teaches wherein said selected one of said virtual threads and said another one of said virtual threads are proxy or interface threads on different computers, the entering of said message in said message queue including transmitting said message over a communications link between said computers *(col. 8, lines 28-31)*.

15. As to claim 11, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Bitar teaches using an interpreter program *(col. 1, lines 26-35)*.

16. As to claim 12, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Bitar teaches a native thread that a virtual thread is mapped from *(col. 13, lines 30-49)*.

17. As to claims 13, Bitar and Diepstraten fails to explicitly teach determining an average load of all the native threads and shifting a virtual thread from a first native thread having a

heavier-than-average load to a second native thread having a lighter-than-average load. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of calculating the average load and shifting the larger avereage load to a lighter load because it is well known that this balance loading will increase the speed and efficiency.

18. As to claim 15, it is rejected for the same reasons as stated in the rejection of claim 10.

19. As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 10.

20. As to claim 17, it is rejected for the same reasons as stated in the rejection of claims 3 and 8.

21. As to claim 18, Bitar teaches wherein each of said virtual threads additionally includes a thread priority, further comprising automatically consulting the thread priorities in a plurality of said virtual threads to determine relative priorities and varying a sequence of threads in accordance with the determined relative priorities *(col. 10, lines 23-29)*.

22. As to claim 20, Bitar teaches wherein said time slots or processing slots are measured by counting consecutively executed pseudocode instructions, further comprising, for each of a plurality of said time slices or processing slots, terminating the respective time slot or processing

slot upon counting a predetermined number of consécutively executed pseudocode instructions (time quantum expires) *(col. 2, lines 22-34)*.

23. As to claim 31, Bitar teaches a computer having an interpreter for executing a series of bytecode instructions each consisting of a multiplicity of machine code steps, a multitasking method comprising:

for each task of a plurality of tasks to be performed by the computer, using the interpreter to define a respective (mapped) virtual thread *(col. 1, lines 26-35)*;

during each time slice of a series of consecutive time slices, executing bytecode instructions of a respective current thread (choose to schedule an alternate thread) selected from among the virtual threads *(col. 11, lines 1-19, col. 13, lines 22-61)*; and

executing a context switch from one of said virtual threads to another of said virtual threads only after execution of one of said bytecode instructions *(col. 2, lines 22-40)*.

24. Bitar teaches virtual threading and context switching *(col. 1, lines 26-40, col. 13, line 61)* but fails to explicitly teach performing context switching only after completed execution of a currently executing one of said pseudocode instructions. However, Diepstarten teaches time slice context switching to occur at the end of every instruction *(col. 8, lines 42-49 and 63-66)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of time slice context switching to occur at the end of every instruction to the existing system of Bitar because it is preferred to switch states or instructions when the instruction is over/completed *(col. 8, lines 50-66)*.

25. As to claims 32-33 and 35-36, they are rejected for the same reasons as stated in the rejection of claims 2-3 and 5-6.

26. As to claim 37, it is rejected for the same reasons as stated in the rejection of claim 8.

27. As to claim 38, it is rejected for the same reasons as stated in the rejection of claim 10.

28. As to claim 39, it is rejected for the same reasons as stated in the rejection of claim 18.

29. As to claim 40, it is rejected for the same reasons as stated in the rejection of claim 11.

30. As to claim 41, it is rejected for the same reasons as stated in the rejection of claim 31. In addition, Bitar teaches a memory storing state and context data of multiple threads or tasks *(col. 6, lines 9-12 and 43-65, col. 5, lines 34—41, and col. 11, lines 1-20)*.

31. As to claim 42, it is rejected for the same reasons as stated in the rejection of claim 2.

32. As to claim 43, it is rejected for the same reasons as stated in the rejection of claim 3.

33. **Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Diepstraten et al. (hereinafter Diepstraten) (US 6,260,150 B1), and further in view of Blanset et al. (hereinafter Blanset) (US 4,744,048).**

34. As to claim 19, Bitar in view of Diepstraten fails to explicitly teach controlling objects imaged on a computer display and monitoring actuation of keys on a computer keyboard. However, Blanset teaches controlling objects imaged on a computer display and monitoring the input of keys on a keyboard *(col. 13, lines 45-60 and see Abstract)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of controlling objects imaged on a computer display and monitoring the input of keys on a keyboard to the existing system of Bitar and Diepstraten because it would allow for context switching by the user *(col. 1, lines 6-26)*.

35. **Claims 44 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Blanset et al. (hereinafter Blanset) (US 4,744,048).**

36. As to claim 44, Bitar teaches a computer method comprising:

compiling input user source code into bytecode or pseudocode instructions each corresponding to a multiplicity of machine code instructions *(col. 5, lines 32-57)*;

operating an interpreter of said computer to assign computing tasks to respective (mapped) virtual threads, the assigning of said computing tasks to said virtual threads including identifying and storing state and context data for each of said computing tasks *(col. 1, lines 25-35, col. 4, lines 57-67, col. 5, lines 19-25, and col. 6, lines 9-12)*;

additionally operating said interpreter to execute selected ones of said bytecode or pseudocode instructions pursuant to the state and context data of a current one of said virtual threads *(col. 11, lines 1-19)*;

during a current one of said time slices, after the execution of each successive one of the selected bytecode or pseudocode instructions and only after such execution, further operating said interpreter to check whether the current one of said time slices has elapsed or terminated since a commencement of execution of instructions pursuant to said current one of said virtual threads *(col. 2, lines 22-34)*; and

upon a determination of elapsing of said current one of said time slices, operating said interpreter to perform a context switch (choose to schedule an alternate thread for some period of time) *(col. 2, lines 35-40)*.

37. As stated above, Bitar teaches operating said interpreter to execute selected ones of said byte- or pseudocode instructions pursuant to the state and context data of a current one of said virtual threads *(col. 11, lines 1-19)* but Bitar fails to explicitly teach that there is a timer to generate a series of time slices to perform this. However, Blanset teaches that timers that keep track of the timing of the time slices *(col. 13, lines 48-52)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of timers because it increases the control of the system *(col. 13, lines 42-59)*.

38. As to claim 45, Bitar teaches the tasks assigned to respective ones of said virtual threads, a network, and calculating local variables *(col. 1, lines 14-35, col. 3, line 50, col. 8, line 30)*. Blanset teaches (a) controlling objects appearing in an image on a display screen, (b) monitoring operator input, (c) executing routines of applications programs, (d) running computer maintenance routines *(col. 13, lines 45-60)*.

Allowable Subject Matter

39. Claim 21-30 are allowed.

40. Claims 4, 14, and 34 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

41. During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." *In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). Applicant always has the opportunity to amend the claims during prosecution, and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. *In re Prater*, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969).

42. Applicant argues in the Remarks regarding claims 1, 11, 12, and 20 that Bitar does not teach using pseudocode instructions.

In response, the Examiner respectfully disagrees. Bitar discloses pseudocode instructions such as "jump instruction" and "load register instruction", for example, which contain programming code related to its respective pseudocode *(col. 15, lines 1-8)*.

43. Applicant argues in the Remarks regarding claim 3 that Bitar does not discuss assigning virtual user-space threads to different kernel-space threads.

In response, the Examiner respectfully disagrees. Bitar teaches assigning logical user-space threads to a resource *(col. 8, lines 34-37)*.

44. *Applicant argues in the Remarks regarding claim 8 that Bitar doesn't teach a message queue.*

In response, Bitar teaches that all communication between kernels is done through messenging *(col. 17, lines 25-26)*. The broadest reasonable interpretation of a message queue is simply a plurality of messages stored in memory.

45. Applicant argues in the Remarks regarding claim 10 that Bitar doesn't teach a proxy or interface thread.

In response, Bitar has an interconnect network 56 that is used to transmit its messenging *(col. 8, lines 28-37)*. It is inherent that there are interface threads within that communication.

46. Applicant argues in the Remarks regarding claims 11-12 that Bitar doesn't teach a virtual resource being directly mapped to a physical resource.

In response, the Examiner respectfully disagrees. Bitar teaches that a logical user space is mapped across a physical resource *(col. 8, lines 27-37)*.

47. Applicant argues in the Remarks regarding claim 19 that Bitar doesn't teach assigning a separate virtual thread to each and every on-screen object.

In response, Bitar teaches assigning a logical kernel space and a logical user space to represent all of physical shared memories *(col. 8, lines 27-37)*.

48. Applicant argues in the Remarks regarding claims 31 and 44 that Bitar doesn't teach an interpreter and bytecode instructions.

In response, Bitar teaches that there is mapping performed between physical and virtual, and the performer of this mapping is the interpreter *(col. 8, lines 27-37)*. It is inherent that bytecode instructions are used because they are simply the computer code instructions for performing computer processes.

49. Applicant argues in the Remarks regarding claim 44 that hat Bitar never mentions compilers or interpreters.

In response, Bitar teaches using compilers (col. 1, lines 15-25). It is also inherent that a computer program has to be compiled before execution.

Conclusion

50. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US 6,233,599 B1 (Nation et al.) discloses a task management system with thread switching after execution is completed (see Abstract).

51. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kt
8/15/05

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100